ChromSeg: Two-Stage Framework for Overlapping Chromosome Segmentation and Reconstruction

Xu Cao*, Fangzhou Lan†, Chi-Man Liu‡, Tak-Wah Lam‡ and Ruibang Luo‡§
*School of Data Science, Fudan University, Shanghai, China
†Department of Computer Science and Technology, Nanjing University, Nanjing, China
‡Department of Computer Science, The University of Hong Kong, Hong Kong, China
§Corresponding author, Email: rbluo@cs.hku.hk

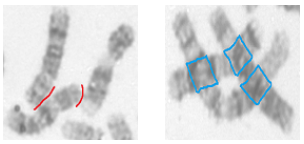
***Abstract*—Karyotyping is the most commonly used genetic tool for diagnosing diseases associated with chromosomal abnormalities. It generates images of the chromosomes of a patient in which quantity or shape discrepancies against normal chromosomes might suggest chromosomal abnormalities. However, the current methods are cumbersome and require manual or half-automatic separation of overlapping chromosomes, significantly limiting the productivity of clinical geneticists and cytologists. In this project, we implemented a fully automatic method, called ChromSeg, which efficiently separates crossing-overlap chromosomes. It uses a new neural network architecture called "region-guided UNet++" to accurately detect crossing-overlap chromosomes from metaphase cell images. A new heuristic algorithm, called "crossing-partition", is then applied to splice and reconstruct the crossing-overlap chromosomes into single chromosomes. While there are a very limited number of publicly accessible annotations on overlapping chromosomes, we manually annotated 345 images for our model training and performance testing. Benchmarking results showed that our method achieved 99.1% overlap detection on crossing-overlap chromosomes and outperformed the second best method by 3.1%. Notably, this is the first tool to provide an image of the reconstructed chromosomes; other tools provide only segmentation suggestions, which are of less value to end-users. The source code of ChromSeg is available at https://github.com/HKU-BAL/ChromSeg, and the 345 annotated images are available at http://www.bio8.cs.hku.hk/bibm/.**

I. Introduction

Karyotyping is the characterization of the number, form and size of chromosomes in an individual's cells. It involves three main steps: 1) capturing metaphase cell images using a microscope camera; 2) applying segmentation on the metaphase cell images to obtain images of individual chromosomes; and 3) classifying and counting the chromosomes[2]. For a single patient, an experienced technician can finish the three steps in a few hours and generate a karyogram with sorted chromosomes, in which structural abnormalities can be identified by a clinical geneticist or cytologist for diagnosing genetic diseases, such as Down's syndrome, Edward's syndrome and Patau's syndrome.

Despite the clinical importance of karyotyping, it is commonly known to be laborious and fallible. The chromosomes are often disentangled in multiple ways in the captured metaphase cell images, so in the second and third step of karyotyping, meticulous manual cutting to segment the chromosomes is almost always needed. It is not uncommon for medical professionals to make mistakes in cutting the entangled chromosomes, resulting in misdiagnosis, especially when the number of samples overwhelms processing power. Thus, computer-aided or fully-automatic methods have been actively explored in the recent years to increase the processing power.

The third step, namely chromosome classification, has been amply facilitated using deep learning-driven methods[16]. However, there are few effective solutions for chromosome segmentation, and some complications in the second step remain unsolved. In Figure 1, we provide two examples of overlapping chromosomes commonly found in a metaphase cell image: one is "touching-overlap" and the other is "crossing-overlap"[7]. While touching-overlap is reportedly solved by recent algorithmic breakthroughs, there is as yet no effective solution for crossing-overlap, which comprise 25.55% of real-life scenarios (calculation explained in the method section).



(a) (b)

Fig. 1: Two examples of overlapping chromosomes. (a) Red lines denote the touching-overlap regions; (b) blue boxes denote the crossing-overlap regions.

Previous attempts to solve overlapping chromosomes have used thresholding[7], morphological analysis[21], [11], and geometric methods[2], [15] to analyze the cutting points[8] and convex hulls[14], which rely heavily on handcrafted features. Each of these algorithms can handle a few testing cases well but fails occasionally in real-life cases, in which exceptions and noise can easily cause confusion. In recent years, researchers have leveraged the power of deep-learning models to better tolerate exceptions and noise. Sharma et al.[19] were the first to use a deep-learning model for karyotyping. They achieved 86.7% accuracy for classification, but for segmentation, their work still relied on traditional algorithms and manual curations from a crowdsourcing platform. Altinsoy et al.[1] applied a U-Net-based neural network for semantic segmentation on metaphase cell images to separate the chromosomes from the background. Their framework achieved an intersection-over-union (IOU) score of 94.11% on 40 metaphase cell images, and

their results were better than all previous thresholding methods. Hu et al.[6] improved the model to produce four masks to separate the non-overlapping parts and overlapping parts of two overlapping chromosomes, resulting in a 94.70% IOU score for overlapping region segmentation. However, Hu's model cannot handle clusters with three or more chromosomes. Very recently, object detection was adapted to segment chromosomes. The idea is to define different chromosomes as different objects, and then leverage the power of general object detection deep-learning models to extract individual chromosomes from a cluster. Li et al. developed DeepACE[22], based on the Faster R-CNN[17] model for automated chromosome enumeration in metaphase images. While only the bounding box instead of the contour of the enumerated chromosomes is provided in DeepACE, the IOU score was not provided for a direct performance comparison with previous methods. Ding et al.[5] worked on a similar idea and added classification to the model. Luo et al.[13] extended DeepACE to accept prior knowledge for segmentation. Compared to the semantic-segmentation methods, object-detection methods localize and separate "touching overlap" chromosomes more effectively, but underperform in handling crossing-overlap chromosomes. Figure 2 shows two examples in which the overlapping chromosomes could not be effectively separated by bounding boxes, so the subsequent classification step failed. In summary, none of the previous methods handle crossing-overlap chromosomes well.

(a) (b)

Fig. 2: Two examples that failed segmentation generated by DeepACE.

In this study, we present ChromSeg, a two-stage deep-learning-based segmentation method, which focuses on maximizing the performance of segmenting and splicing crossing-overlap chromosomes. ChromSeg takes crossing-overlap images from DeepACE or Faster R-CNN as input. In the first stage of ChromSeg, it uses a multi-layer aggregation semantic segmentation convolutional neural network, based on deep-layer aggregation (DLA)[23] and UNet++[24] to localize the input image's crossing regions and separate chromosomes from the background. The input image is divided into crossing-overlap regions and non-overlap regions in the first stage. In the second stage, all crossing-overlap regions and non-overlap regions are divided and reconstructed using a novel heuristic algorithm, named "crossing-partition". The output of ChromSeg comprises single chromosome images segmented from the input. From 535 metaphase cell images, which is the highest among all the previous methods mentioned above, we extracted 345 crossing-overlap images that failed segmentation in DeepACE and annotated them manually for model training and performance testing. The benchmarks show ChromSeg outperformed all vision methods in crossing-overlap chromosome segmentation.

Our main contributions are summarized as follows:

- We designed and implemented ChromSeg, a two-stage deep-learning based segmentation method, which focuses on maximizing the performance of segmenting crossing-overlap chromosomes.
- We manually annotated 345 "cross overlapping" images that failed segmentation using existing tools for model training and performance testing. The images are freely accessible by other research teams.
- Compared to the second-best method, ChromSeg increased the segmentation accuracy of crossing-overlap chromosomes from 96.0% to 99.1%.
- This is the first tool to provide an image of the reconstructed chromosomes; other existing tools provide only segmentation suggestions, which are of less value to end-users.

The next section provides a detailed description of our dataset and the ChromSeg method. The heuristic algorithm "crossing-partition" is described in pseudo-code. Section 3 presents our benchmarking results and ablation studies and draws conclusions from them. Section 4 discusses a possible defect in ChromSeg and discusses our future research directions. Our conclusions are provided in the final section.

II. Method

In this section, we introduce a new crossing-overlap chromosome annotation dataset with 345 images and our two-stage ChromSeg model. ChromSeg is a two-stage framework, based on the divide-and-conquer concept. In the first stage of ChromSeg, we use a deep-learning method to pinpoint and divide overlapping regions in the input images. In the second stage, we use improved geometry and union-find methods to predict and splice crossing-overlap and non-overlap regions. The output is individual chromosomes that can be successfully spliced. The model introduction comprises three parts: 1) the architecture of the region-guided UNet++, 2) the Mixed Weight Focal (MWF) loss function, and 3) a crossing-partition algorithm. The framework of ChromSeg is shown in Figure 3.

A. New crossing-overlap chromosome annotation dataset

There are a few chromosome annotation datasets from previous studies for solving the automatic karyotyping problem. The best are Hu's fluorescent chromosome dataset[6] for semantic segmentation and DeepACE's Peking University Third Hospital metaphase cell images dataset[22] for object detection. However, as explained in the introduction, none of the previous studies and their annotations effectively separated crossing-overlap chromosomes. To quantify the impact of crossing-overlap chromosomes, we summarized different numbers of overlapping chromosomes from 535 metaphase cell images from the University of Hong Kong. The statistics are shown in Table I. If we assume that each metaphase cell image contains 46 chromosomes, i.e. 24,610 chromosomes in 535 images,

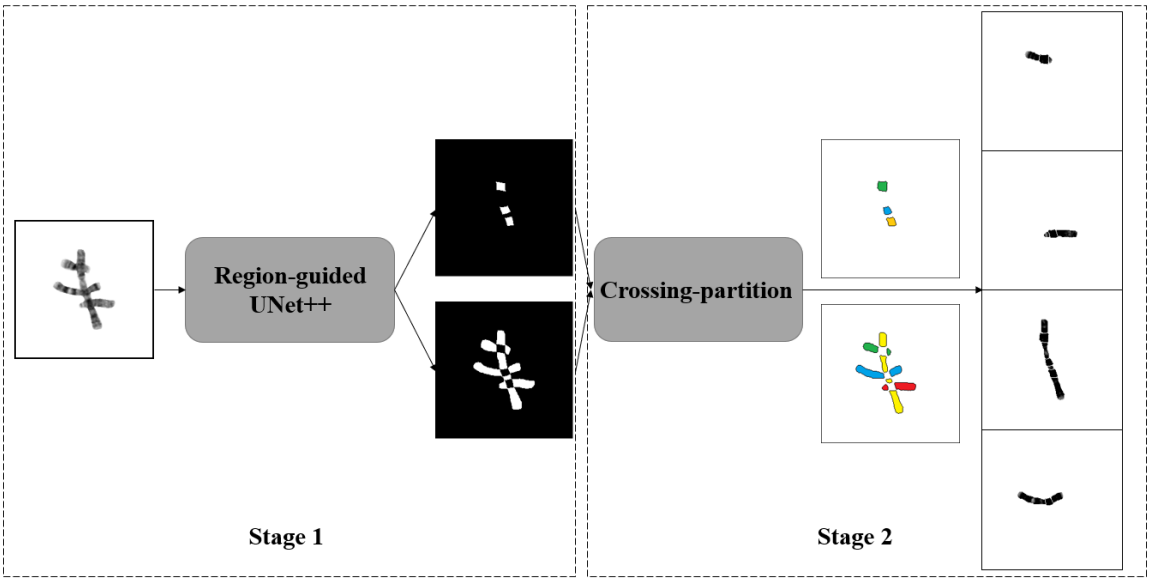


Fig. 3: ChromSeg framework.

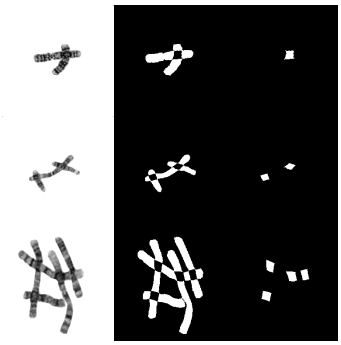


Fig. 4: Our annotation for crossing-overlap chromosomes. Each original image (left) is annotated with two binary images representing the crossing-overlap regions (middle) and the non-overlap regions (right).

TABLE I: Statistics of cross-overlap chromosomes from 535 real-life metaphase cell images. The percentages are calculated as "(# of overlapping chromosome * count) / 24,610".

Overlapping chromsomes	Count
2	1,460 (11.86%)
3	514 (6.27%)
4	220 (3.58%)
5 or more	189 (3.84%)
All overlapping chromosomes	6,287 (25.55%)

the ratio of crossing-overlap chromosomes is approximately 25.55%.

In order to solve the crossing-overlap chromosome segmentation problem, we selected and annotated 345 crossing-overlap chromosome images of size 256×256. All of the images were annotated with two binary images representing the crossing-overlap regions and the non-overlap regions. An example is shown in Figure 4. As annotation is time-consuming and laborious, we selected even-number representative images of different numbers of overlapping chromosomes. It is true that for deep-learning applications, the more training samples the better. Nevertheless, later in our experiments, we show that 345 annotated images are enough for good performance. The dataset is available at http://www.bio8.cs.hku.hk/bibm/.

B. Region-guided UNet++

Figure 5 illustrates the architecture of the first stage, i.e. the region-guided UNet++. This is a semantic segmentation network for separating crossing-overlap regions and non-overlap regions of input images. The idea for the region-guided UNet++ was inspired by UNet++, which is a convolutional neural network for image segmentation that adds densely aggregation-modules over its predecessor U-Net's skip connection to fuse information better[23]. UNet++ and its sister framework, DLA, were demonstrated to be better than U-Net on cell medical images[9], [3], [24].

To separate overlapping regions of metaphase cell images, two problems must be solved: 1) separating the background and objects, and 2) pinpointing crossing-overlap regions. Theoretically, multiple semantic segmentation networks can be used to handle different parts of the input in parallel. However, the size of a good-performing semantic segmentation network, UNet++[24], DeepLab series[4] for example, are not small. Using more networks might increase the sensitivity, but it will also increase the computation linearly, which is not practical. To solve this problem, we reconsidered the characteristics of the two problems and found a simple and effective solution.

In our region-guided UNet++, we extended UNet++ to be a multi-branch region predictor. The input of the region-guided UNet++ is a 256×256 RGB 3-channel metaphase cell image. We used a shallow U-Net (depth 4) as the encoder-decoder backbone and added three middle convolution layers to the skip

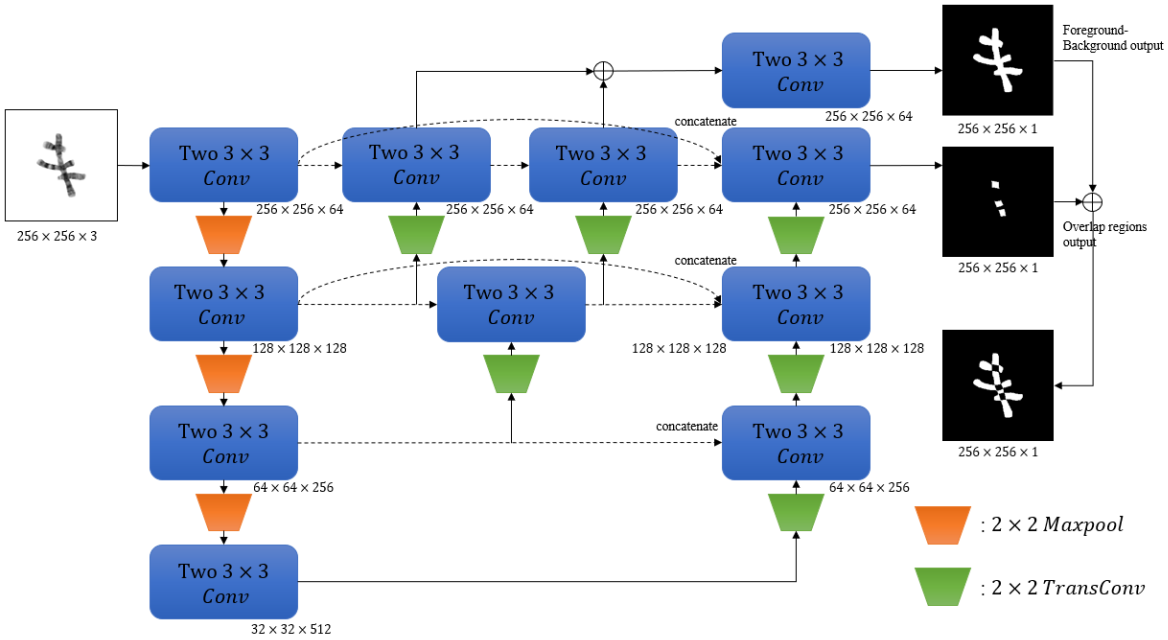


Fig. 5: Structure of the region-guided UNet++. The output of the region-guided UNet++ contains two probability maps: 1) foreground-background (top-right), and 2) overlap regions (middle-right). The third probability map, non-overlap chromosome regions (bottom-right), is generated from the other two probability maps.

connection path. This structure can iteratively and hierarchically fuse information across different levels of features, and its output can be optimized by deep supervision and pruning approaches. In addition, we extended UNet++ to a two-branch dual-output framework, producing a foreground-background output and a crossing-overlap region output. The rationale of a two-branch output is to utilize the features at multiple levels to solve the two problems simultaneously. As shown in Figure 5, the network utilizes shallow and adjacent spatial features to separate the background from the foreground, and aggregates all information from multiple layers to predict the crossing-overlap regions. Our region-guided UNet++ generates two probability maps of size W×H×C, with 1) P_f representing the foreground-background, and 2) P_c representing the overlap regions, where W is the width and H is the height of the input image, and the channel dimension C is 1 for each probability map.

Our experiments show that the region-guided UNet++ solved the two problems simultaneously and effectively. It achieved a better IOU score than the existing U-Net using equal numbers of parameters.

C. Mixed-Weight Focal (MWF) loss function

At the end of the region-guided UNet++, two 1×1 convolution layers with sigmoid activation function are used to extract final features and produce output probability maps of foreground-background and crossing-overlap regions. In these two probability maps, the foreground objects, including the individual chromosomes and crossing-overlap regions, occupy only a small fraction of pixels, meaning that our crossing-overlap region segmentation task is an imbalanced pixel-wise classification problem. With imbalanced classes, training a network with the standard cross-entropy loss will fail early in the network training (falling into a local minimum) or lead to degenerate models with poor performance.

We designed a mixed-weight focal loss function to solve the imbalanced classification problem. Focal loss is applied to trade off positive and negative sample training[12]. The focal loss is defined as:

$$FL(P_t) = \begin{cases} -\alpha_t(1-P_t)^{\gamma} log(P_t) & y=1 \\ -(1-\alpha_t)P_t^{\gamma} log(1-P_t) & y=0 \end{cases} \quad (1)$$

, where P_t is the predicted probability, y is the ground truth, 1 denotes a positive sample, and 0 denotes a negative sample. α_t and γ are parameters that control class balance. α_t is a weighting factor between the positive and negative samples. γ is used to control the steepness and smoothness of the weight distribution. When γ=0, the focal loss is equivalent to cross-entropy loss. Higher weight is applied to the minority class with increasing γ.

In the region-guided UNet++, the two probability maps outputs are trained against their corresponding truth annotation. Therefore, a combined-loss function with discounted weights was used to train this network. The Mixed-Weight Focal (MWF) loss defined below worked effectively in the region-guided UNet++.

$$\begin{aligned} &L_f(P_f) = FL(P_f) \qquad L_c(P_c) = FL(P_c) \\ &MWFloss(P_f, P_c) = \lambda L_c(P_c) + (1-\lambda)L_f(P_f) + \phi \end{aligned} \quad (2)$$

, where L_f and L_c are the focal loss of the two probability maps; λ is a hyperparameter for balancing the trade-off between two terms; and ϕ is a regularization term. Experiments using different hyperparameters in the loss function are shown in the hyperparameter study subsection in Section 3.

D. Crossing-partition algorithm

The crossing-partition algorithm extracts single chromosomes from the candidate crossing-overlap regions detected by the region-guided UNet++. First, the watershed algorithm[20] is used to divide the overlapping chromosomes in a crossing-overlap region into two groups: overlapping parts and non-overlapping parts (see stage 2 in Figure 3). The target of the crossing-partition algorithm is to group the parts from the two groups accurately to form individual chromosomes. As shown in Algorithm 1 (Function Build Dictionary), we identify the non-overlapping parts that are supposed to be connected with an overlapping part by searching in a small disk area of the overlapping part. The disk area is a dilation of the overlapping part. The disk area is set to contain the overlapping part and all of its authentic non-overlapping parts as much as possible. We set the dilation of the disk area at about 8 to 12 pixels (See DiskAreaSize d in Algorithm 1). It is rare, if not impossible, for an overlapping part to belong to three or more chromosomes. In real-life applications, the problem is usually solved by using another metaphase cell image of the same cell taken at a different time, so our algorithm doesn't handle overlapping parts that consist of more than two chromosomes. That being said, each disk area of an overlapping part should contain no more than four adjacent non-overlapping parts that belong to exactly two chromosomes, forming shapes including the "X" shape, "T" shape and "L" shape. If we can properly pair up the non-overlapping parts, we can then use a union-find algorithm to efficiently assign the parts to individual chromosomes.

In order to properly pair-up the non-overlapping parts, we designed a heuristic algorithm. The pseudo code of the algorithm is shown in Algorithm 1 (Function Classify). For each overlapping part and its non-overlapping parts, the algorithm first calculates the central coordinate of all the parts. Specifically, when calculating the central coordinate of the non-overlapping parts, the algorithm considers only the points within the disk area of the overlapping part it belongs to in order to avoid being affected by chromosomes with unexpected shapes or large curliness. Then a straight line is connected from each centre of the non-overlapping parts to the centre of the crossing-overlap part, and then extended past the centre to scan other parts (like emitting a ray from the non-overlapping parts to the overlapping part). If an extended straight line intersects with at least one non-overlapping part in the disk area, we conclude that the two parts belong to the same chromosome. Using this algorithm, we found that most of the complicated crossing-overlap chromosomes cases can be effectively segmented and reconstructed (see Figure 6).

After using Function Classify to group all non-overlapping parts, we obtained a forest based on the union-find algorithm. Then we traversed all trees in the forest and combined

Algorithm 1 Crossing-partition

Input: OverlapBlock V: V_0,V_1,...,V_{g-1}
NonoverlapBlock S: S_0,S_1,...,S_{k-1}
DiskAreaSize: d
Output: Chromosome C: C_0,C_1,...

function BUILD DICTIONARY(V, S, d)
  for V_i in V **do**
    for S_j in S **do**
      if S_j adjacent to V_i in scope d **then**
        $D[V_i]$.Add(S_j);
      end if
    end for
  end for
  Return D;
end function

function CLASSIFY(V, D, d)
  Init Forest F;
  for V_i in V **do**
    x_i, $y_i \leftarrow$ Find Center(V_i);
    for S_j in $D[V_i]$ **do**
      x_j, $y_j \leftarrow$ Find Center(S_j);
      $S_h \leftarrow$ Ray Emit((x_i, y_i),(x_j, y_j)) in scope d;
      if S_h is not $NULL$ **then**
        Add S_h to F(UnionFind(S_j));
      end if
    end for
  end for
  Return F;
end function

$D \leftarrow$ Build Dictionary(V, S, d);
$F \leftarrow$ Classify(V, D, d);
for $Tree$ in F **do**
  C_n.Add($Tree$.Root);
  for V_i in V **do**
    if V_i adjacent to $Tree$.Root **then**
      C_n.Add(V_i);
    end if
  end for
  for Each Node in $Tree$.Root **do**
    C_n.Add(Node);
    for V_i in V **do**
      if V_i adjacent to Node and V_i not in C_n **then**
        C_n.Add(V_i);
      end if
    end for
  end for
end for
Return C;

them with crossing-overlap parts to output all individual chromosomes.

III. Experiments and results

To test ChromSeg and prove its robustness, we trained a region-guided UNet++ on our newly labeled ChromSeg dataset and tested the crossing-partition algorithm on different types of crossing-overlap chromosome images. We carried out an ablation study and a hyperparameters study for our proposed two-stage method.

A. Dataset and training details

There were 345 crossing-overlap images in our dataset. The size of each image was 3×256×256 pixels, and its ground truth was two binary masks annotating crossing-overlap regions and chromosome foreground regions. We divide the dataset into 230 training images and 115 test images. The training data was augmented using 30-degree rotation and random horizontal and vertical flipping.

The region-guided UNet++ was implemented in PyTorch. An Adam optimizer[10] was used to optimize models with the learning rate set to 1e-4. The batch size was set to 8 for all training data. The default hyperparameter of Mixed Weight Focal loss was λ=0.75, α=0.75, γ=1. More details are provided in Subsection D. Early-stopping on the validation set was applied with a patience of 5 epochs. The proposed network was trained on three NVIDIA GTX 1080Ti GPU.

B. Evaluation metrics

The performance of the first stage region-guided UNet++ in our experiment was measured by two metrics: mean intersection over union (IOU or Jaccard's index) and overlap detection accuracy (OverlapAcc). The performance of the second stage (crossing-partition algorithm) was measured by splicing accuracy (SPAcc). The two accuracies are defined as:

$$OverlapAcc = \frac{N_{wd}}{N} \times 100\% \tag{3}$$

$$SPAcc = \frac{N_{ws}}{N_{wd}} \times 100\% \tag{4}$$

, where N_{wd} indicates well-detected crossing-overlap images in the first stage, N_{ws} indicates well-spliced images in the second stage, and N indicates the total number of test images. The overall accuracy for crossing-overlap chromosome segmentation is counted as OverlapAcc×SPAcc.

C. Comparison and ablation study

First, we compared our proposed method with other methods. The best-reported vision approach for detecting crossing-overlap regions is the computational geometry method, based on the cut-point. Since most of the state-of-art methods reported in Section 1, including DeepACE, do not deal with the separation of crossing-overlap chromosomes, our ChromSeg is the first deep-learning method that tries to solve this problem. Table II shows the performance comparison between our proposed method and the computational geometry method.

TABLE II: Performance comparison between ChromSeg and other methods.

Method	IOU (crossing-overlap)	IOU (foreground)	OverlapAcc	SPAcc
Computational geometry[8]	-	-	96.0%	-
U-Net[18]	78.7%	-	98.2%	-
ChromSeg	**80.8%**	**99.7%**	**99.1%**	**91.3%**

The authors of the computational geometry approach, based on the cut-points, mentioned that their method worked well on 2 or 3 crossing-overlap cases, but errors still occurred when dealing with more complicated crossing-overlap cases. Since most of the images in their dataset had 2 or 3 crossing-overlap chromosomes, their experiments proved that the geometry approach could effectively separate most of simple overlapping cases[8]. But they did not propose an approach to splice and reconstruct every individual chromosome. In our experiment, we found that ChromSeg performed very well in handling all simple and complex crossing-overlap cases, as the region-guided UNet++ of ChromSeg improved the overlap-detection accuracy from 96.0% to 99.1%, including all crossing-overlap cases. On the basis of region-guided UNet++'s detected crossing-overlap regions, we used the crossing-partition algorithm to splice chromosome segments for the first time. Figure 6 shows the excellent segmentation and reconstruction results of the crossing-partition algorithm. It achieved accuracy of 91.3% in separating all individual chromosomes in different complicated cases. After integrating the two stages, ChromSeg achieved overall accuracy of 90.5% when it localized and separated every individual chromosome in crossing-overlap chromosome clusters. We also compared the first stage of ChromSeg with U-Net, the most commonly used semantic segmentation method, in the medical image as an ablation study. In the experiment, the performance of region-guided UNet++ (IOU and overlap detection accuracy) was better than U-Net by 1.9% IOU (See Table II).

The comparison experiments proved that ChromSeg is the best solution for crossing-overlap chromosome segmentation and reconstruction. The performance of our method might be improved further by expanding the image data in the future.

D. Hyperparameters study

Based on the experiments in Table II, we studied the selection of hyperparameters in the first stage. Mixed Weight Focal (MWF) loss has huge advantages in training models. It can trade off the training process between two probability maps and also allows the model to focus on learning imbalanced positive samples instead of being distracted by easily learned negative samples. However, the hyperparameters (λ,α,γ) in MWF loss can have a direct effect on the results of the experiment, and setting hyperparameters empirically does not allow the model to perform well most of the time. Thus, we designed the hyperparameters study to select the best hyperparameters in MWF loss. Table III shows the hyperparameters study for α, γ in MWF loss. Another hyperparameter, λ, was selected to a

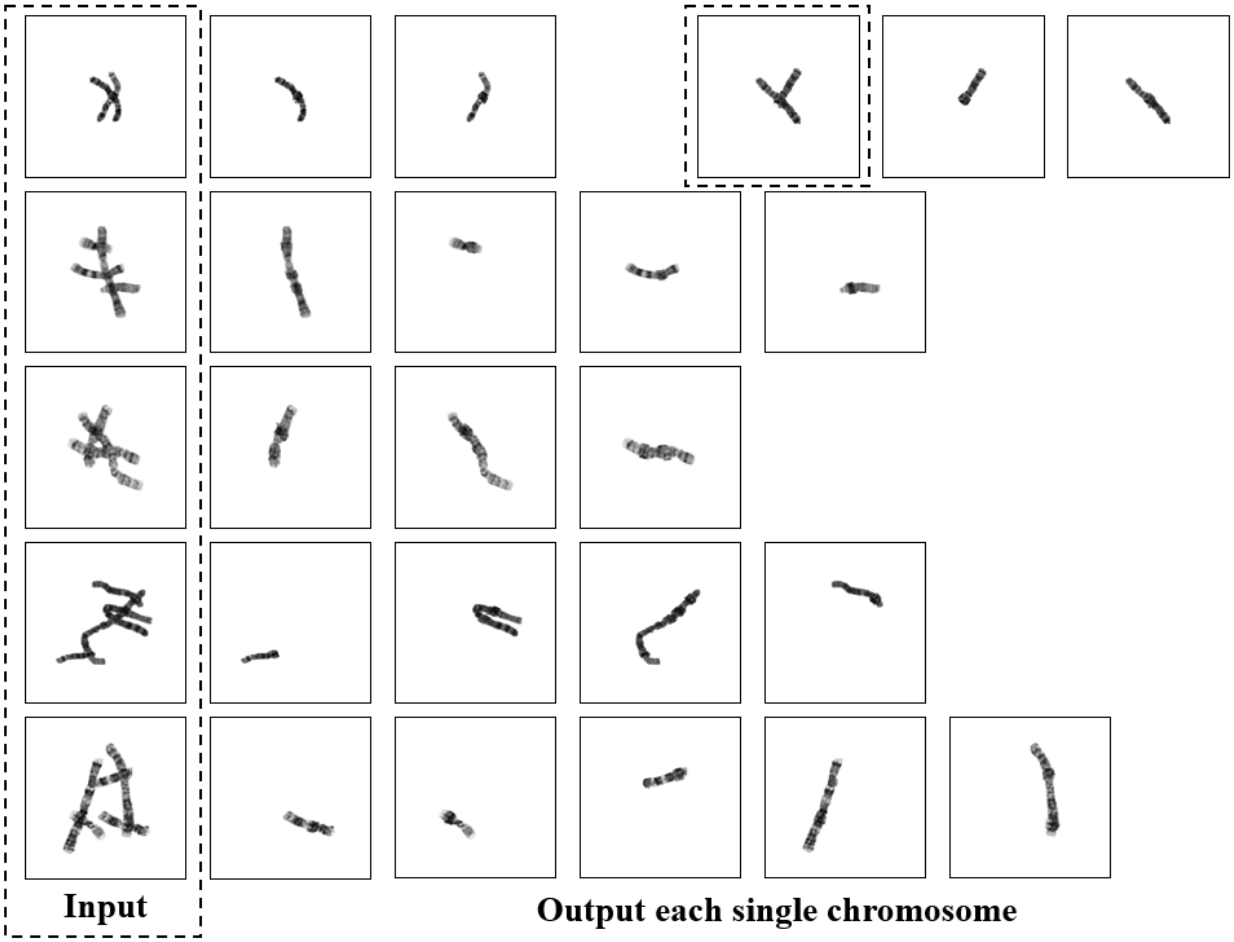


Fig. 6: The segmentation result of ChromSeg. It handles crossing-overlap situations including the "X" shape, "L" shape, "T" shape, and a mix of the above shapes.

TABLE III: Hyperparameters findings for α, γ. γ=0, α=0.5 denotes the binary cross entropy loss.

γ	α	IOU (crossing-overlap)	IOU (foreground)
0	0.5	79.98%	99.70%
1	0.25	69.64%	99.11%
	0.50	79.36%	99.71%
	0.75	**80.62%**	**99.74%**
	0.90	75.90%	99.69%
2	0.25	58.80%	98.74%
	0.50	73.76%	99.66%
	0.75	79.80%	99.70%
	0.90	77.92%	99.71%

fixed value, 0.5, in this experiment. The findings indicated that γ=1, α=0.75 is the best hyperparameter for detecting crossing-overlap regions. The hyperparameters were not sensitive to detecting chromosome foreground regions. Table IV shows the hyperparameters study on λ, indicating that λ=0.7 is the best hyperparameter for training region-guided UNet++.

IV. Discussion

The results show that ChromSeg can solve most of the complex cases of crossing-overlap chromosome segmentation and reconstruction. However, there are still some cases that ChromSeg fails to solve. In this section, we evaluate these errors and offer future direction for solving them.

TABLE IV: Hyperparameters experiment result for λ. γ=1, α=0.75.

λ	IOU (crossing-overlap)	IOU (foreground)
0.9	78.85%	99.75%
0.7	**80.77%**	99.75%
0.5	80.62%	99.74%
0.3	79.96%	99.75%
0.1	79.09%	99.75%

The failed cases that we found in the entire two-stage segmentation and reconstruction task can be divided into 3 classes: 1) segmentation error in crossing-overlap region detection (stage 1); 2) splicing error due to the connection between the non-overlapping parts (stage 2); and 3) two or more overlapping regions being too close, causing connecting crossing-overlap (stage 2). We can solve the first case by expending the dataset. For the other two cases, however, there is still a lack of suitable solutions. We find that these two types of failed cases have a feature in common: two parts have

(a) (b)

Fig. 7: Two examples of failed cases. (a): blue boxes denote connecting crossing-overlap; (b): red circles denote connecting non-overlapping parts.

shared connecting areas. They can be seen as a combination of touching-overlap and crossing-overlap problems. A possible solution is to design a multi-branch or multi-task ChromSeg + Fast R-CNN framework. Before running ChromSeg, we could first use Fast R-CNN[17] or other methods for localizing to remove touching-overlap regions that would affect the segmentation.

V. Conclusion

Karyotyping is a core topic in medical image analysis and bioinformatics because of its usefulness in genetic diagnosis. In this paper, we focused mainly on solving the challenging task of crossing-overlap chromosome segmentation and reconstruction. To accomplish precise segmentation, we proposed a novel two-stage method and annotated a new dataset for crossing-overlap chromosome segmentation. In the first stage of the model, our region-guided UNet++ and MWF loss achieved 99.1% overlap detection accuracy in separating crossing-overlap regions. In the second stage of the model, the crossing-partition algorithm completed the chromosome image reconstruction and output for each chromosome for the first time. In summary, ChromSeg achieved state-of-the-art performance, with 90.5% overall segmentation and reconstruction accuracy. This model has huge potential to be combined with Fast R-CNN-based object detection methods to achieve automatic karyotyping.

Acknowledgment

R. L. was supported by the ECS (grant number 27204518) of the HKSAR government, and by the URC fund at HKU.

References

[1] Emrecan Altinsoy, Can Yilmaz, Juan Wen, Lingqian Wu, Jie Yang, and Yuemin Zhu. Raw g-band chromosome image segmentation using u-net based neural network. In *International Conference on Artificial Intelligence and Soft Computing*, pages 117–126. Springer, 2019.

[2] Tanvi Arora and Renu Dhir. A novel approach for segmentation of human metaphase chromosome images using region based active contours. *International Arab Journal of Information Technology*, 16(1):132–137, 2019.

[3] Xu Cao and Yanghao Lin. Caggnet: Crossing aggregation network for medical image segmentation. *arXiv preprint arXiv:2004.08237*, 2020.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[5] Wei Ding, Ling Chang, Chaochen Gu, and Kaijie Wu. Classification of chromosome karyotype based on faster-rcnn with the segmatation and enhancement preprocessing model. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5. IEEE, 2019.

[6] R Lily Hu, Jeremy Karnowski, Ross Fadely, and Jean-Patrick Pommier. Image segmentation to distinguish between overlapping human chromosomes. *arXiv preprint arXiv:1712.07639*, 2017.

[7] Liang Ji. Fully automatic chromosome segmentation. *Cytometry: The Journal of the International Society for Analytical Cytology*, 17(3):196–208, 1994.

[8] Mukul A Joshi, Mousami V Munot, Madhuri A Joshi, Kruti R Shah, and Ketan Soni. Automated detection of the cut-points for the separation of overlapping chromosomes. In *2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences*, pages 820–825. IEEE, 2012.

[9] Qingbo Kang, Qicheng Lao, and Thomas Fevens. Nuclei segmentation in histopathological images using two-stage learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 703–711. Springer, 2019.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Boaz Lerner. Toward a completely automatic neural-network-based human chromosome analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(4):544–552, 1998.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[13] Chunlong Luo, Tianqi Yu, Yufan Luo, Manqing Wang, Fuhai Yu, Yinhao Li, Chan Tian, Jie Qiao, and Li Xiao. Deepacc: Automate chromosome classification based on metaphase images using deep learning framework fused with prior knowledge. *arXiv preprint arXiv:2006.15528*, 2020.

[14] Nirmala Madian and KB Jayanthi. Overlapped chromosome segmentation and separation of touching chromosome for automated chromosome classification. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5392–5395. IEEE, 2012.

[15] Shervin Minaee, Mehran Fotouhi, and Babak Hossein Khalaj. A geometric approach to fully automatic chromosome segmentation. In *2014 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6. IEEE, 2014.

[16] Yulei Qin, Juan Wen, Hao Zheng, Xiaolin Huang, Jie Yang, Ning Song, Yue-Min Zhu, Lingqian Wu, and Guang-Zhong Yang. Varifocal-net: A chromosome classification approach using deep convolutional networks. *IEEE Transactions on Medical Imaging*, 38(11):2569–2581, 2019.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[19] Monika Sharma, Oindrila Saha, Anand Sriraman, Ramya Hebbalaguppe, Lovekesh Vig, and Shirish Karande. Crowdsourcing for chromosome segmentation and deep classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–41, 2017.

[20] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):583–598, 1991.

[21] Yan Wenzhong and Li Dongming. Segmentation of chromosome images by mathematical morphology. In *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, pages 1030–1033. IEEE, 2013.

[22] Li Xiao, Chunlong Luo, Tianqi Yu, Yufan Luo, Manqing Wang, Fuhai Yu, Yinhao Li, Chan Tian, and Jie Qiao. Deepacev2: Automated chromosome enumeration in metaphase cell images using deep convolutional neural networks. *IEEE Transactions on Medical Imaging*, 2020.

[23] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[24] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2019.